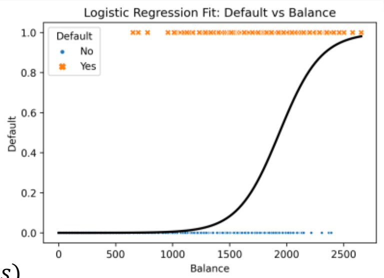
Cost based Evaluation
Cross-validation and Regularization

BA810: Supervised Machine Learning

Nachiketa Sahoo

Recap

- Logistic regression
  - learn log odd ratio as a linear function
- Bayesian approaches
  - Use Bayes rule to invert the direction of conditional probability
  - Compute $\Pr(class|attribute)$ from $\Pr(attribute|class)$ and $\Pr(class)$
  - I.e., posterior distribution of class from class conditional distribution and prior
- K-nearest neighbor
  - Select k most similar points to the test point from training data
  - Predict majority label (classification) or mean (regression)
- Evaluation
  - Confusion matrix, True Positive, True Negative, False Positive, False Negative
  - Accuracy, Precision, Recall, F-measure, ROC curve



Outline

- Cost based evaluation and prediction
  - Use expected cost of classification to choose classifier given costs
  - Cost minimizing prediction
- Cross validation
- Regularization

Cost Based Evaluation and Prediction

Cost Matrix

Because different *types* of mistakes could have different costs. Examples?

	PREDICTED CLASS		
ACTUAL CLASS	C(i|j)	**Class=Yes**	**Class=No**
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

C(i|j): Cost of misclassifying class j example as class i

Computing Cost of Classification

+ : fraudulent transaction

Cost Matrix	PREDICTED CLASS		
	C(i|j)	+	-
ACTUAL CLASS	+	-1	100
	-	1	0

Model M_1	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	150	40
	-	60	250

Accuracy = 80%

Cost = 3910

Model M_2	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	250	45
	-	5	200

Accuracy = 90%

Cost = 4255

Confusion Matrices

Tuning Prediction to Minimize Cost

- The cost of False Positive is \$12, cost of False Negative is \$3
  - If you are using a classifier that estimates the probability of a record being +ve, what is the lowest probability at which you'd classify a record to be +ve?

Hint: look for the probability threshold where the expected cost from mistake by either prediction is same. At that probability you'll be indifferent between predicting a record to be + or -

Record#	P(+)	Predict
1	0.99	+
2	0.91	+
3	0.84	?
4	0.75	?
	…	…
19	0.23	?
20	0.11	-

Cross Validation

Estimating Generalization/Test Error

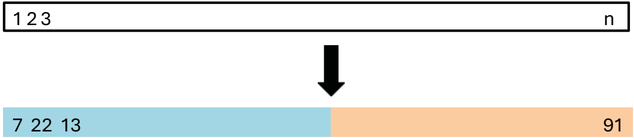
Two Approaches

to estimate generalization error

- Hold out a subset (validation set)
  - Learn the model from the remainder
  - Evaluate on validation set
    - MSE for regression, misclassification rate for classifier

- Mathematical adjustment to the training error rate to estimate the test error rate
  - Penalize training performance for complexity/number of parameters
  - Cp statistic, AIC, BIC, adjusted R^2
  - Less accurate, but useful when repeated training and testing is costly (e.g., feature selection)

The Validation Process

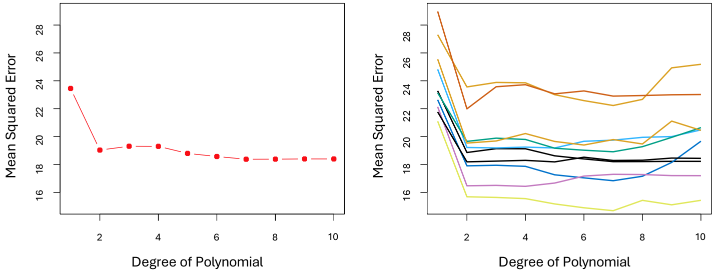


Randomly split into two parts: training and validation sets.

A large portion of the dataset is kept for validation, so that our new data error estimates are reliable.

Example: automobile data

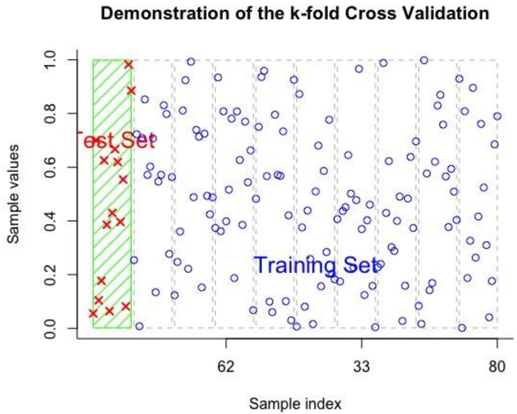
- To compare linear vs higher-order polynomial regression
- Randomly split the 392 observations into two halves: one for training, one for validation
  - Fit to training, measure MSE on validation, plot for different degrees of polynomials



Left panel shows single splitting; right shows multiple splitting

K-fold Cross-validation

- Randomly divide the data (with n records) into K equal parts
  - leave out k'th part for validation
  - fit the model to the other $K - 1$ parts (combined)
  - measure performance on the left-out k'th part
- Repeat for each $k = 1, 2, ... K$, and take weighted average of the metrics
  - Setting $K = n$ is called leave-one-out-cross-validation



Source: https://yihui.org/animation/example/cv-ani/

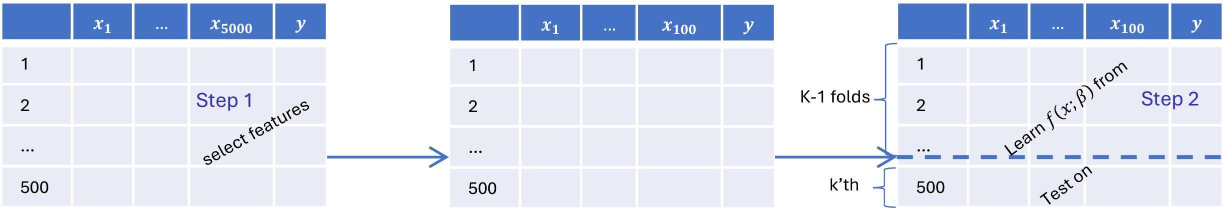
Cross-validation: Right And Wrong Way

Consider learning a regression model for a dataset with 5000 features and 500 rows.

Standard Multiple Linear regression can’t be estimated with more features than records. So,

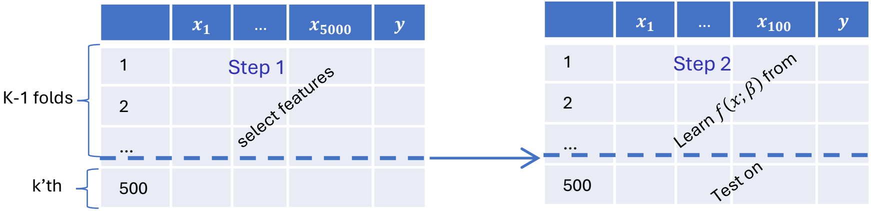
1. One selects the 100 features most correlated with the target.
2. Then learn a linear regression using only these 100 features.

- How do we estimate the test set performance of this classifier?
    - Can we wrap step 2 in cross validation to measure test error, after step 1?



NO!

- This would ignore that in Step 1, the procedure has already used the labels of the training data — an example of data leakage, from test to training.
  - Ignoring the use of all data in feature selection underestimates the test data error.
  - Feature selection is a form of training and must be separated in the cross-validation process.
- Right: Wrap both steps 1 and 2 in cross validation.



Regularization

Simplifying models in a controlled way

Incurring some bias to reduce variance and overall prediction error

Two Classes of Methods to Control Complexity

- Regularization/Shrinkage. Fit a model involving all predictors, but the estimated coefficients are shrunken towards zero (relative to the least squares estimates).
  - Reduces variance and some can perform variable selection.

- Feature Selection. Identify a subset of the predictors that are most related to the response. Then fit a model using least squares on the reduced set of variables.

Regularization

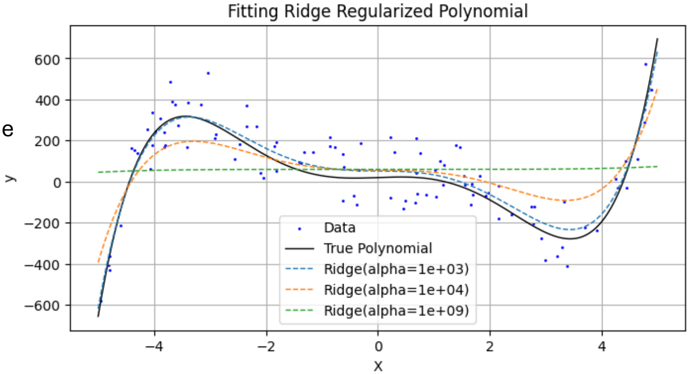
Minimize a modified objective instead of MSE on training data

Ridge Regression Minimizes:

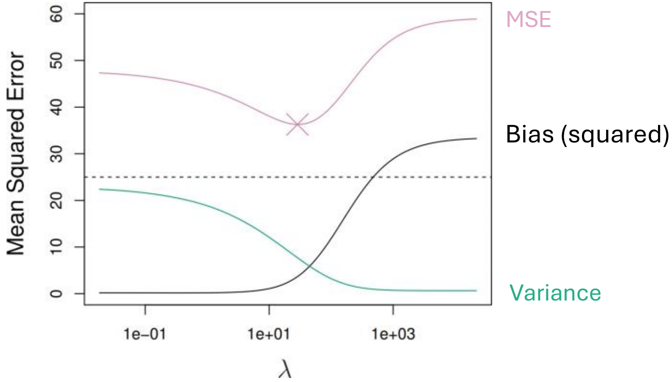
$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- First term: sum of squared residuals
- Second: penalizes large coefficients
  - Which can result from fitting to noise in the training data
- $\lambda > 0$ is a tuning parameter, often chosen by cross validation

- Larger λ shrink the coefficients more
  - But all βs remain non-zero, i.e., doesn't select features.
  - Important to standardize the features first



Bias–Variance Tradeoff with Regularization Parameter λ



Recall that MSE = Variance + Bias + Irreducible error

Source: Tibshirani et al.

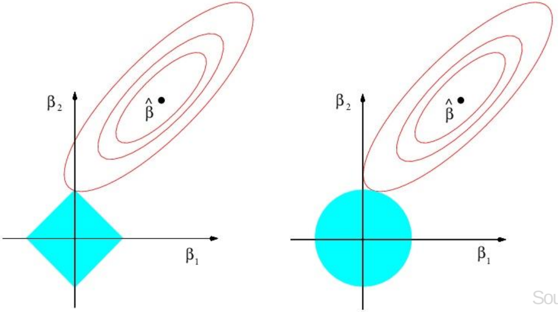
Lasso Regression

- Lasso (Least Absolute Shrinkage and Selection Operator) minimizes:

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

This turns coefficients, one-by-one, to exactly zero as λ increases → selects features

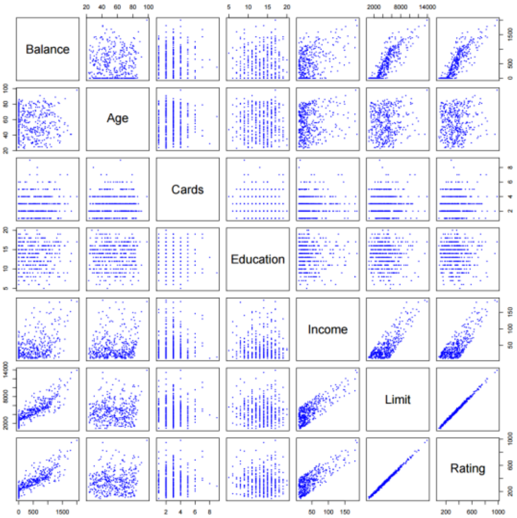
Lasso



Ridge

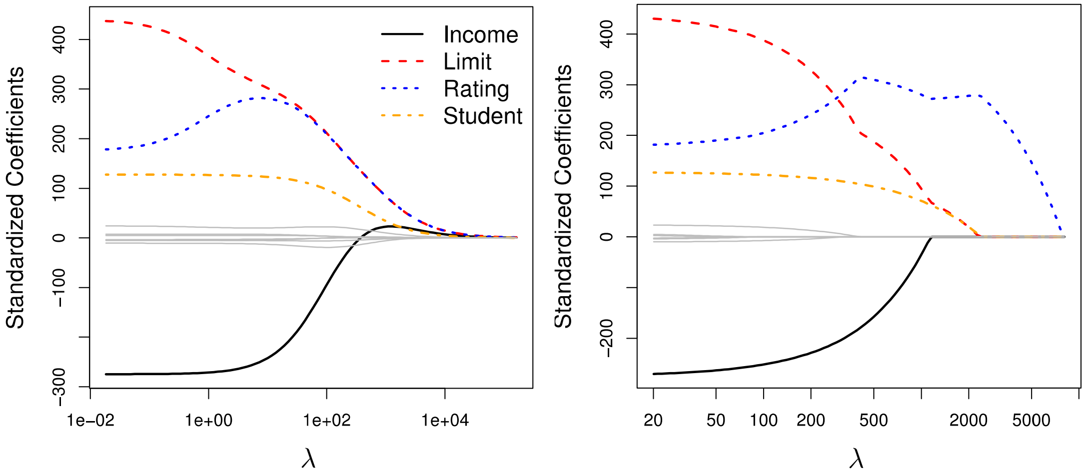
Source: Tibshirani et al.

Example: Predicting Credit Card Balances



Source: Tibshirani et al.

Ridge vs Lasso in Predicting Credit Card Balance



Source: Tibshirani et al.

Ridge or Lasso?

- If there truly are only a few relevant variables (rest are noise), then Lasso is likely to perform better
  - Often not known in advance
  - Use cross validation to choose among Lasso and Ridge (as well as the regularization parameter)
- To choose regularization parameter λ, for each method
  - Choose a set of regularization parameters λs
  - Using each λ, fit k models to cross validation training data, average test errors
  - Choose λ with lowest test error
  - Retrain the model using the chosen λ and entire training data

Summary

- Cost based evaluation
  - Choose the model with the least expected cost of prediction
- Cost minimizing prediction
  - Change the positive/negative threshold to minimize the expected cost out of mistakes
- Cross validation
  - Use 1 of K parts for evaluation using the rest for training (K times)
  - Can't use the same validation data to select a model and to measure its generalization error
  - Cross validate within the training data to select model
- Regularization to control complexity
  - Reduce variance, potentially incur bias
- In linear regression
  - Ridge: reduces coefficients towards zero
  - Lasso: reduces coefficients towards zero and can turn some to exactly zero
    - Can be used for feature selection
- Many types of regularization depending on forms of predictive models
  - k (# of neighbors) in kNN regularizes; high k reduces variance, potentially inducing bias